



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

OPTIMIZING FORMATION MOVEMENT OVER HETEREGENEOUS TERRAIN

by

Fatih CESUR

June 2005

Thesis Advisor:
Second Reader:

Gerald G. Brown
Ellen F. Roland

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE June 2005	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE: Optimizing Formation Movement Over Heterogeneous Terrain			5. FUNDING NUMBERS	
6. AUTHOR(S) Fatih Cesur				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) Formation movement is vital to preserve security among its units during military operations. We plan movement of a military formation over real, or simulated terrain, maximally preserving the relative positions of units in formation while it avoids barriers, and while its units avoid obstacles. Terrain is divided into homogeneous cells (say, squares), and a pair of neighboring cells is adjacent if the formation can transit between these cells while avoiding barriers with sufficient clearance. We induce a graph from these adjacencies, and determine the movement cost on each arc with a fine time-step simulation that finds local movement vectors to preserve relative formation position while avoiding approach too close to barriers or obstacles (this emulates solving differential equations with Euler's method). We then nominate an origin and a destination, select a shortest path, and repeat the time-step simulation over this path to determine the individual positions of each unit as the formation makes its transit. Game designers and robot controllers have published schemes to guide formation movement, but their movements can penetrate barriers, and myopically get caught in cul-de-sacs. By contrast, we guarantee that if a path exists that avoids these pitfalls, we will find it.				
14. SUBJECT TERMS Formation Movement, Shortest Path, Dijkstra, Tactical Movement, Maintain Formation			15. NUMBER OF PAGES 59	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**OPTIMIZING FORMATION MOVEMENT OVER HETEREGENEOUS
TERRAIN**

Fatih Cesur
First Lieutenant, Turkish Army
B.S., Turkish Military Academy, 2000

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

**NAVAL POSTGRADUATE SCHOOL
June 2005**

Author: Fatih Cesur

Approved by: Gerald G. Brown
Thesis Advisor

Ellen F. Roland
Second Reader

James N. Eagle
Chairman, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Formation movement is vital to preserve security among its units during military operations. We plan movement of a military formation over real, or simulated terrain, maximally preserving the relative positions of units in formation while it avoids barriers, and while its units avoid obstacles. Terrain is divided into homogeneous cells (say, squares), and a pair of neighboring cells is adjacent if the formation can transit between these cells while avoiding barriers with sufficient clearance. We induce a graph from these adjacencies, and determine the movement cost on each arc with a fine time-step simulation that finds local movement vectors to preserve relative formation position while avoiding approach too close to barriers or obstacles (this emulates solving differential equations with Euler's method). We then nominate an origin and a destination, select a shortest path, and repeat the time-step simulation over this path to determine the individual positions of each unit as the formation makes its transit. Game designers and robot controllers have published schemes to guide formation movement, but their movements can penetrate barriers, and myopically get caught in cul-de-sacs. By contrast, we guarantee that if a path exists that avoids these pitfalls, we will find it.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	MOTIVATION	1
B.	PURPOSE.....	2
II.	LITERATURE REVIEW	3
A.	RELATED STUDIES	3
1.	Balch and Hybinette [2000].....	3
2.	Kamphuis and Overmars [2004]	6
3.	Crombie [1997].....	6
4.	Aragon [2001]	7
B.	MOVEMENT FORMATIONS.....	8
III.	FORMATION MOVEMENT.....	11
A.	CONSTRUCTING THE TERRAIN NETWORK.....	11
B.	FORMATION MOVEMENT IN THE NETWORK	12
1.	Move to a Goal Location	15
2.	Maintain Formation.....	15
3.	Avoid Barriers and Obstacles.....	17
4.	Formulation	17
a.	<i>Index Use</i>	18
b.	<i>Data</i>	18
5.	A Solution to Computational Complexity.....	19
C.	FINDING ARC COSTS FOR THE NETWORK	21
D.	SHORTEST PATH-FINDING ALGORITHM	23
IV.	MODEL IMPLEMENTATION	27
A.	MOVEMENT SCENARIO.....	27
B.	RUNNING THE MODEL.....	28
C.	MAZE PATH-FINDING.....	33
V.	CONCLUSIONS	37
	LIST OF REFERENCES.....	39
	INITIAL DISTRIBUTION LIST	41

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1.	Attachment sites: Four different formations (diamond, line, column and square) with different numbers of attachment sites (From Balch and Hybinette, [2000]).	3
Figure 2.	A force field to avoid a static obstacle.	4
Figure 3.	The magnitude of avoid a static obstacle vector as a function of distance r .	4
Figure 4.	Force field computation to maintain formation, move to goal or move to unit center parameters.	5
Figure 5.	The magnitude of maintain formation, move to goal or move to unit center vector as a function of distance r .	5
Figure 6.	Agent movement constraint (From Aragon, [2001]).	7
Figure 7.	Column formation.	9
Figure 8.	Line formation.	9
Figure 9.	Echelon formation.	9
Figure 10.	Vee formation.	10
Figure 11.	Wedge formation.	10
Figure 12.	A sample 3x3 terrain layout with 9 nodes and adjacencies to neighboring nodes.	11
Figure 13.	A sample 3x3 terrain layout with 6 nodes and 2 barriers.	12
Figure 14.	A sample grid terrain layout with barriers and obstacles.	13
Figure 15.	An instance of formation movement of a unit from node s to node t .	14
Figure 16.	Parameters for move to a goal location and maintain formation rules.	15
Figure 17.	Attachment sites.	16
Figure 18.	Parameters for avoid obstacles rule.	17
Figure 19.	Running time as a function of number of units with obstacles and barriers.	20
Figure 20.	Running time as a function of number of units without obstacles or barriers.	21
Figure 21.	Time-step simulation to determine travel distance from one node to an adjacent node while following a barrier or obstacle repulsion function.	22
Figure 22.	A display to calculate arc costs as a node-to-node time step simulation.	23
Figure 23.	The effect of obstacles on arc cost from adjacent node s to node t .	24
Figure 24.	A sample 2x3 terrain network with arc costs.	25
Figure 25.	Forward star representation of a sample data set (After Ahuja, <i>et al</i> , [1993, pp. 35-37]).	25
Figure 26.	Dijkstra's algorithm using a 2-heap (From Ahuja, <i>et al</i> , [1993, p. 115]).	26
Figure 27.	AOR for the mechanized infantry platoon.	27
Figure 28.	Shortest path from platoon assembly area to check point 3.	29
Figure 29.	Example of appearance of distorted formation.	30
Figure 30.	Shortest path length as a function of barrier effective range.	31
Figure 31.	Shortest path length as a function of obstacle effective range.	32
Figure 32.	Shortest path length versus barrier effective range and obstacle effective range.	33
Figure 33.	Maze terrain.	34

Figure 34.	The shortest path from s to t.	34
Figure 35.	Maze terrain with no path.	35
Figure 36.	Output window of the Java application.	35

LIST OF TABLES

Table 1.	Running time versus number of units with obstacles and barriers.	19
Table 2.	Running time versus number of units without obstacles or barriers.....	19
Table 3.	A comparison of shortest path lengths with various barrier effective ranges.	30
Table 4.	A comparison of shortest path lengths with various obstacle effective ranges.	31
Table 5.	A comparison of shortest path lengths with various barrier and obstacle effective ranges.	32

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to express deep gratitude to my country and the Turkish Armed Forces for this opportunity.

I express sincere appreciation to Professor Brown for his guidance, assistance and patience throughout the research. I am thankful to Mrs. Roland for her advice and support on the topic.

I express my thanks and appreciation to my family for their understanding, and patience. I am also thankful to 1LT Grabner, LTC Brau and his wife Jennifer, MAJ Fale, Miss Olson, Jean and Mike LaPorte, Robin and Chris Grabner and all my colleagues and friends who made my stay at the NPS and Monterey a memorable and valuable experience.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

We develop a new method to plan tactical movement of military formations either over real terrain, or in a combat simulation.

Coordinated movement of subordinate units is vital for tactical movement planning. Currently, there are two types of coordinated formation movement used in the real world and represented in military simulations: administrative movement and tactical movement.

The primary concern of administrative and tactical movement is to follow a secure route that enhances integrity of formation units. The critical areas that might increase the alert status of a formation while moving toward its goal location are mountains, rivers, or lakes that might disrupt coordination of units. There are also some areas that might complicate unit movements such as a group of trees, urban areas, or small hills. The concept to movement planning is to keep a secure distance between these critical areas and units.

To optimize formation movement over a simulated terrain that carries the characteristics of a real terrain, we divide terrain into 2-dimensional homogeneous grid cells, create barriers to movement that a formation must always move around, and obstacles that a formation can split up and pass around on both sides. We define a repulsion function for each barrier and obstacle that units in the formation will try to avoid. We also define attraction functions for goal locations that each unit must approach. We induce a graph between grid cells if the formation can transit between these cells while it avoids barriers, and while its units avoid obstacles. Movement costs between adjacent grid cells within the graph are calculated using a time step simulation that finds local movement vectors to preserve relative formation position while avoiding approach too close to barriers or obstacles. We then nominate an origin and a destination, select a shortest path, and repeat the time-step simulation over this path to determine the individual positions of each unit as the formation makes its transit.

We prove that if a path exists from an origin to a destination in our terrain, we will find it. The proof is constructive: we find every adjacency that can accommodate formation movement. We then define a graph with all such adjacencies, and apply a well-known path-finding method.

In our scenario, we plan tactical movement for a platoon level formation from its assembly area to three successive check points. Our method finds the shortest path for these units that guarantees maintaining formation. We also demonstrate our method over a maze-like terrain. As long as a path exists, this method finds it.

I. INTRODUCTION

A. MOTIVATION

We develop a new method to plan tactical movement of military formations either over real terrain, or in a combat simulation. There are three main types of tactical operations for a platoon-level formation: movement, offensive operations and defensive operations [U.S. Army, 1992]. Movement achieves suitable conditions for offensive and defensive operations. Movement gains the ability to implement the planned maneuver, followed by the main tactical operation.

Maneuver positions forces in terrain to gain advantage over the enemy [U.S. Army, 1985]. Maneuver is an essential part of tactical operations for every level of unit, from squad to army. Effective tactical movement planning gives the commander the capability to properly execute maneuver in case of contact with the enemy.

Coordinated movement of subordinate units is vital for tactical movement planning. Currently, there are two types of coordinated formation movement used in the real world and represented in military simulations: administrative movement and tactical movement.

Administrative movement is used if there is no possibility of enemy contact and the terrain offers amenable conditions. Tactical movement is adopted when contact with the enemy is possible. The distinguishing difference between administrative and tactical movement is the level of security. While planning movement, the commander should consider the following rules [U.S. Army, 1985]:

- Use the terrain for protection: The movement plan should consider the geographical features of the terrain.
- Avoid possible kill zones: The formation should avoid possible kill zones which provide situational superiority to the enemy.
- Take active countermeasures: A detailed movement plan is itself the best countermeasure.
- Maximize the armored personnel carrier's capability: The movement plan should consider the capabilities of the carrier.
- Make contact with the smallest force possible: There are five main movement formations for a mechanized infantry platoon each of which

provides for different levels of contact with the enemy. While moving across terrain, changing into different formations decreases the vulnerability of the unit against unexpected enemy attacks.

B. PURPOSE

The objective of this thesis is to create a formation movement shortest path-finding algorithm for a military unit to implement tactical unit movement in formation over heterogeneous terrain.

Grid squares will be used to represent heterogeneous terrain features, though any alternate terrain division will serve as well. The grid terrain layout will contain *obstacles* and *barriers* that will complicate coordinated movement planning. Barriers such as mountains, lakes, rivers or oceans are impossible for the formation to cross. Barriers force the formation to move around as a whole. Obstacles, such as small hills, a group of trees, a small swamp or a creek, can make movement difficult, but individual units of the formation can move around obstacles on both sides.

We seek a shortest path that maximally preserves a desired unit formation while advancing in the terrain and avoiding barriers and obstacles that may deform the unit formation, and also avoiding colliding into friendly units.

Finally, we demonstrate tactical movement of a mechanized infantry platoon using a wedge formation over a heterogeneous grid terrain layout containing barriers and obstacles.

II. LITERATURE REVIEW

A. RELATED STUDIES

1. Balch and Hybinette [2000]

This paper describes mobile robot navigation that has an increasing attraction in special operations, reconnaissance operations, and search and rescue missions. The paper introduces new potential functions for a coherent group of robots to maintain their formation while navigating thorough a specially-generated terrain with different obstacles.

These authors create different types of geometric formations using attachment sites, or ideal positions for subordinate units. Every formation has different number of attachment sites depending on the number of units in the formation and its geometric shape. Figure 1 shows different formations and attachment sites.

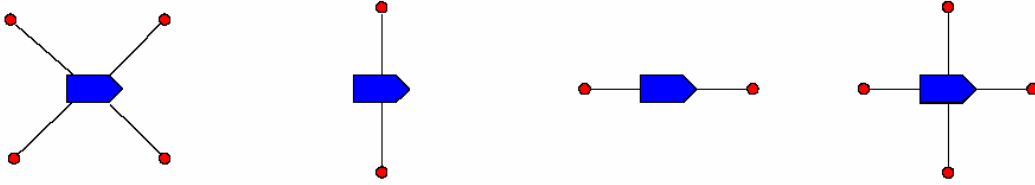


Figure 1. Attachment sites: Four different formations (diamond, line, column and square) with different numbers of attachment sites (From Balch and Hybinette, [2000]). The pentagon represents the formation leader and the short sides indicate the direction of movement. The dots are “attachment sites,” or ideal positions to be occupied by formation units whenever possible. If formation units cannot occupy their sites, then they should assume positions as near as possible.

Robots follow rules while navigating over terrain and maintaining their formation. These rules include “move to goal,” “avoid static obstacles,” “avoid other robots,” “maintain formation,” and “move to unit center”. Attraction and repulsion functions are used to implement these rules while the robots are moving. The parameters used for the avoid static obstacles function are shown in Figures 2 and 3, parameters used for maintain formation, move to goal and move to unit center functions are shown in Figures 4 and 5.

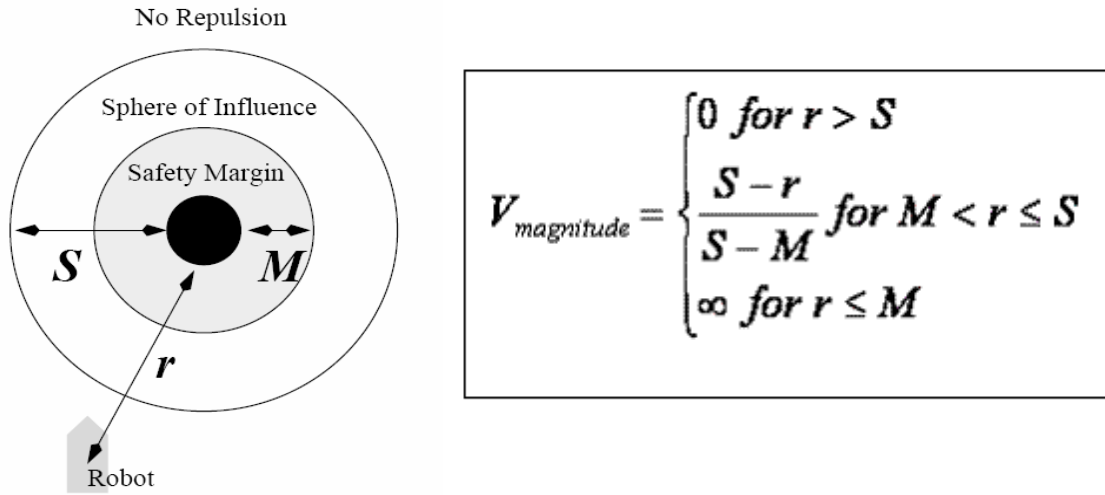


Figure 2. A force field to avoid a static obstacle. Balch and Hybinette, [2000] use S for sphere of influence, M for the safety margin, and the distance of the robot to the obstacle is represented by the letter r . The black point at the center of the inner circle is a static obstacle. The magnitude of the repulsion vector is 0 when the robot is outside the sphere of influence, ∞ when the robot is within the safety margin and $(S-r) / (S-M)$ when $M < r \leq S$. The direction of the repulsion vector is always from the obstacle through the robot.

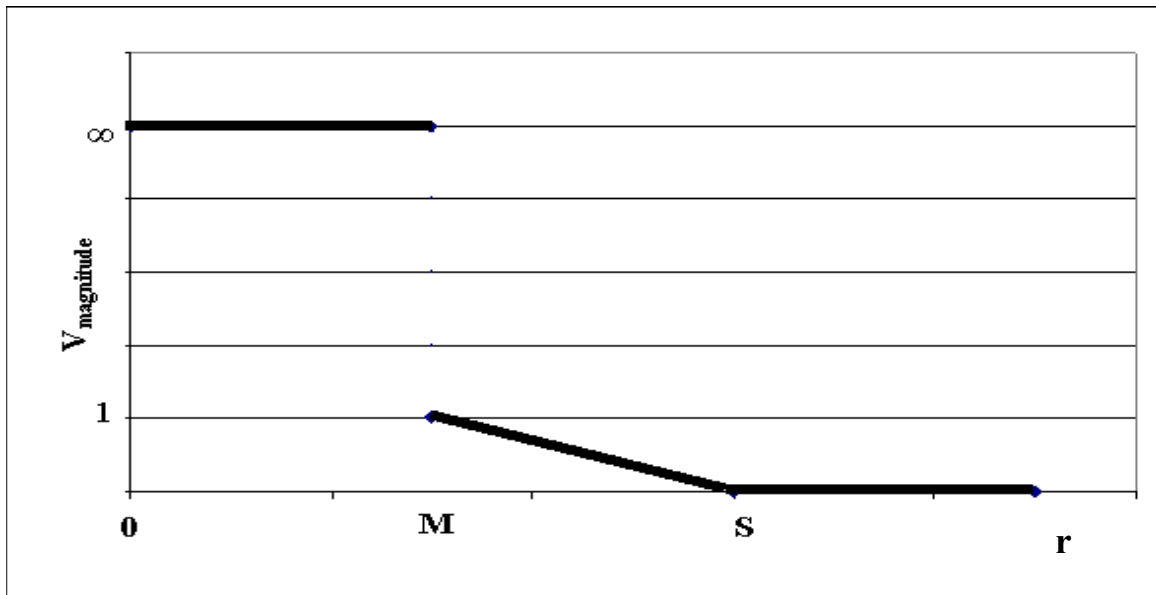


Figure 3. The magnitude of avoid a static obstacle vector as a function of distance r . In this figure, S stands for sphere of influence; M stands for safety margin, r for the distance of the robot to the obstacle. If the robot's distance to obstacle is closer than M , the magnitude of the vector is infinity which repels the robot away from the obstacle.

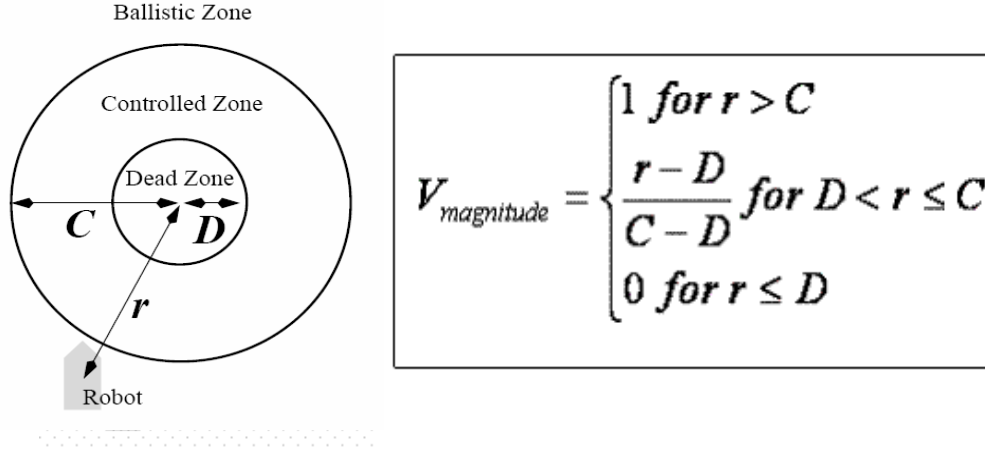


Figure 4. Force field computation to maintain formation, move to goal or move to unit center parameters.

Balch and Hybinette, [2000] use C for the controlled zone, D for the dead zone, and r for the distance of the robot to the goal location. The area outside the controlled zone is called the ballistic zone. The center of the inner circle is the attraction point. The magnitude of the vector is 1 when the robot is in the ballistic zone, 0 when the robot is in the dead zone and $(r-D) / (C-D)$ when $D < r \leq C$. The direction of the vector is always through the goal location.

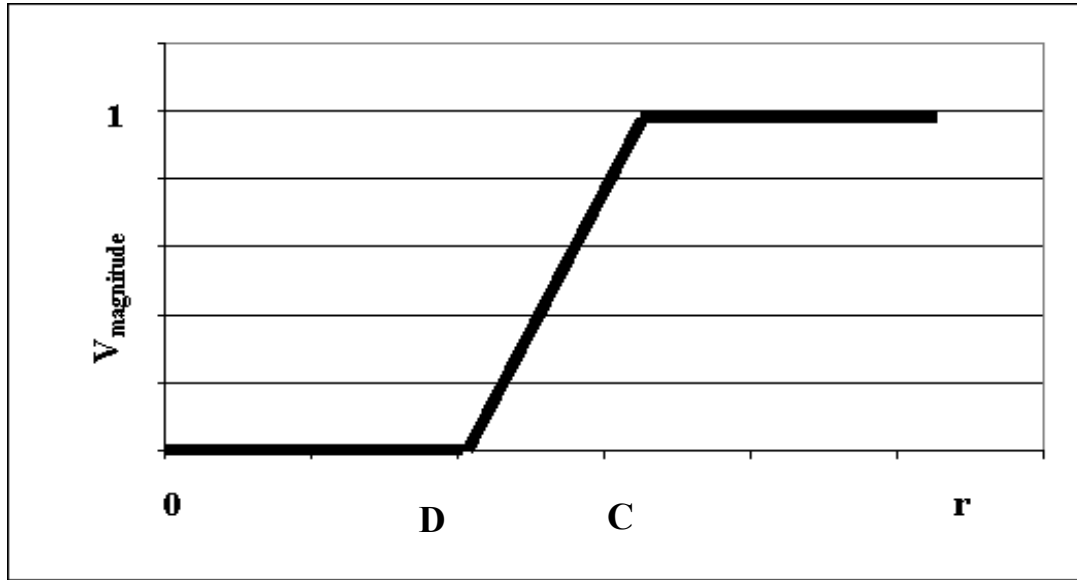


Figure 5. The magnitude of maintain formation, move to goal or move to unit center vector as a function of distance r .

In this figure, C stands for the controlled zone; D stands for dead zone, r for the distance of the robot to the goal location. If the robot's distance to the goal location is more than C , the magnitude of the resultant vector is always 1 which attracts the robot to the goal location.

2. Kamphuis and Overmars [2004]

Kamphuis and Overmars introduce the idea of using clearance to extend the path that is found for a single unit to a corridor path with sufficient clearance to accommodate a coherent formation. With the corridor path, a formation is guaranteed to be able to move as a coherent group inside the corridor along this path.

The objectives are to move a coherent group of units from a starting area to a destination area and to avoid units colliding with each other or with the obstacles in the terrain. These authors first find a path for a single unit. This path has a minimum clearance at every point of the path that allows the group movement along the path. The second step is the implementation of the group movement through this “backbone path”. The distances to obstacles on every point along the backbone path are calculated and the backbone path is extended to a corridor path that maintains a minimum separation from obstacles on every point along the path.

3. Crombie [1997]

Crombie examines different algorithms to control multiple robots that are moving together. He creates a Java [2005] application to simulate and display the movement of the units and analyze the implementation of the algorithms.

He uses three basic rules:

- Cohesion: This rule forces the entities to move toward the average point of all entities.
- Alignment: This rule forces the entities to align their direction to the group's direction.
- Separation: This rule forces the entities to keep a minimum distance from each other and avoids collision between units.

Crombie uses three vectors to move each entity and keep entities in a group; an alignment, attraction and repulsion vector. The alignment vector maintains the direction of the entities, the attraction vector keeps the group together, and the repulsion vector pushes the entities away from obstacles. The repulsion vector also affects the entities when they get closer to each other and avoids collisions. At each increment of simulated time, the successive locations of each entity are calculated by a vector summation of these three vectors in planar coordinates, followed by a movement in the resulting direction.

4. Aragon [2001]

Aragon presents an agent-based simulation of a Marine infantry squad that operates in urban terrain. He creates a Java application with a graphical user interface to demonstrate the motion of the squad conducting a patrolling task through an urban area.

The terrain is generated as a two-dimensional system of grid squares. Each member of the squad can move to eight different directions as shown in Figure 6.

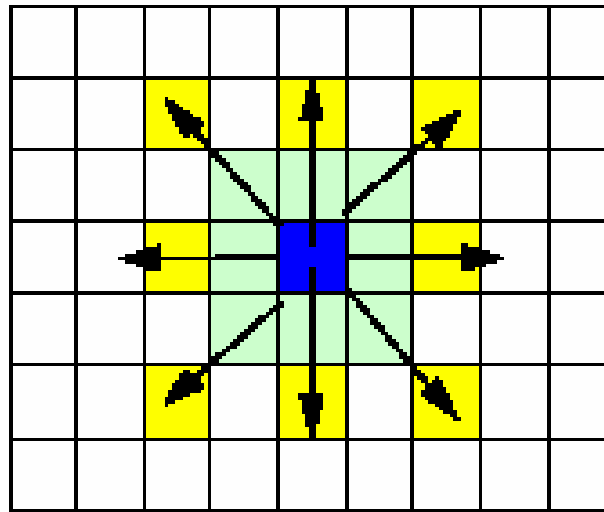


Figure 6. Agent movement constraint (From Aragon, [2001]). The center square represents the location of an agent. Each agent can move to 8 directions shown with arrows.

In the model, only the squad leader is capable of path planning. The squad members follow the squad leader according to their assigned locations in the formation. Aragon presents a collision detection and avoidance method that ensures that the agents do not collide into each other while following their leader. At each time step, the method checks candidate grid squares for each agent to move, identifies if these grid squares are occupied by other agents or buildings. If there is an available square, the agent moves to that square, otherwise it keeps its current location until it finds an available grid square.

B. MOVEMENT FORMATIONS

We focus on tactical movement planning of a mechanized infantry platoon that consists of a platoon leader and three squads equipped with armored personnel carriers (APC). Different movement formations provide unique advantages and disadvantages. The main objective of the leader is to keep the security level as high as possible while moving over the terrain as fast as possible. The platoon leader makes a judgment of the situation in order to decide which movement formation and technique to use. The leader's guide for this quick judgment is known as METT-T factors [U.S. Army, 1985]. These factors are:

- Mission: The platoon leader clarifies the task of the unit with the purpose of the commander.
- Enemy: The platoon leader analyzes the enemy's past, current and future capabilities that may affect the mission.
- Terrain: The platoon leader considers significant effects of the area of operation on the mission.
- Troops: The platoon leader examines the composition of friendly and enemy units.
- Time available: The platoon leader considers the amount of available time on hand to accomplish the task.

When mounted on vehicles, a mechanized infantry platoon has five types of movement formations [U.S. Army, 1985]. These are:

- Column formation (see Figure 7),
- line formation (see Figure 8),
- echelon formation (see Figure 9),
- vee formation (see Figure 10), and
- wedge formation (see Figure 11).

Movement formations are used to establish communication between the units, establish firepower over the enemy, divide areas of responsibility between each subordinate unit, and ease the command and control of all the formation units.

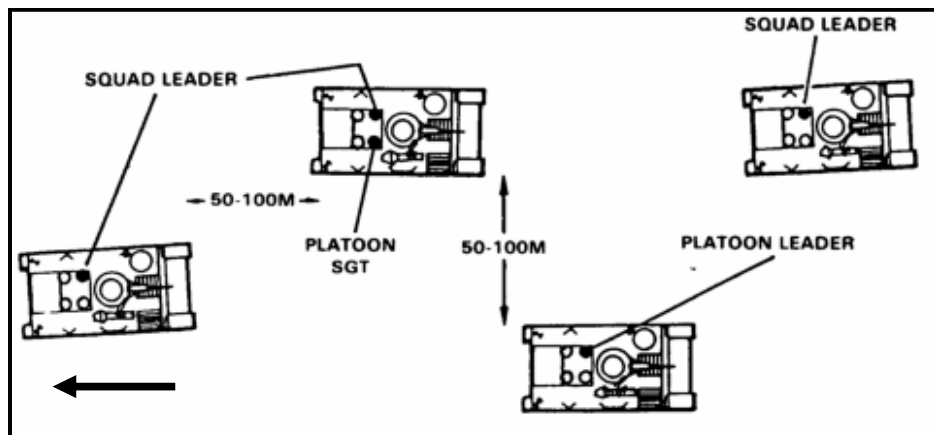


Figure 7. Column formation.
Movement direction is shown with black arrow.

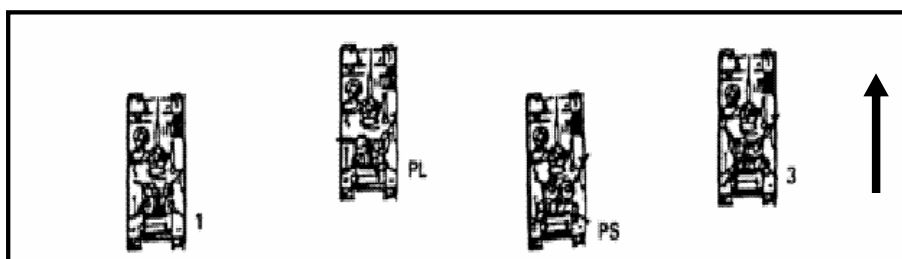


Figure 8. Line formation.

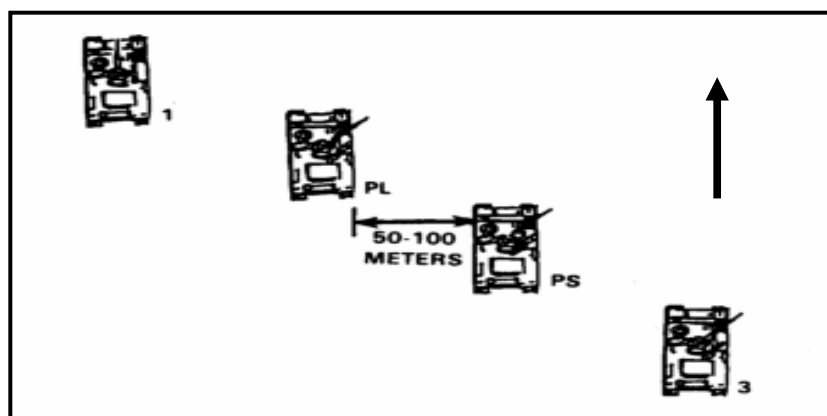


Figure 9. Echelon formation.

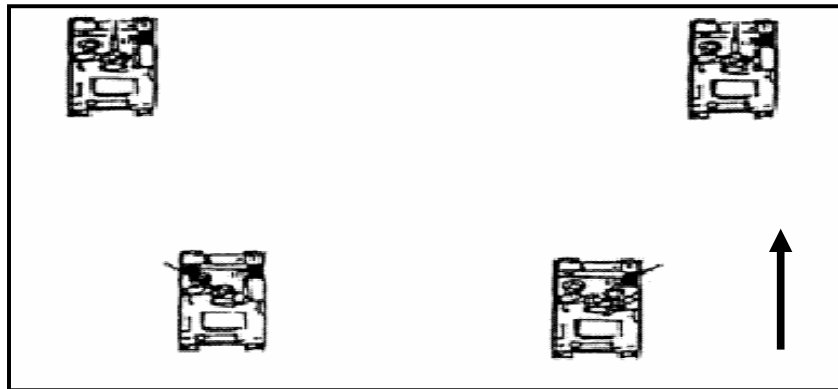


Figure 10. Vee formation.

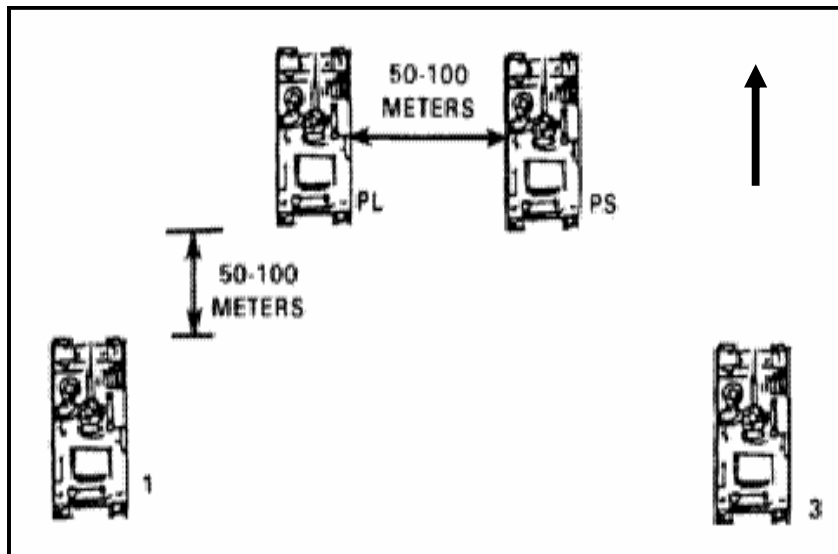


Figure 11. Wedge formation.

The distance between the APCs depends on the terrain, enemy situation and visibility. There are no strict rules on the distance, but the criterion is to maintain communication and be ready to react against any unexpected enemy activity.

III. FORMATION MOVEMENT

A. CONSTRUCTING THE TERRAIN NETWORK

Our terrain layout consists of two-dimensional grid squares that are homogeneous in size. The layout contains 70x50, or 3500 grid squares. Within the terrain layout there are barriers and obstacles.

A barrier is a mountain, a river, or a lake which occupies a large area and prevents formation movements. In our terrain layout some grid squares are “barriers” and formations may not transit such terrain.

Every square that is not a barrier contains a node in its center. Each square can have a maximum of 8 adjacent grid squares (see Figure 12) assuming that there are no barriers within adjacent squares.

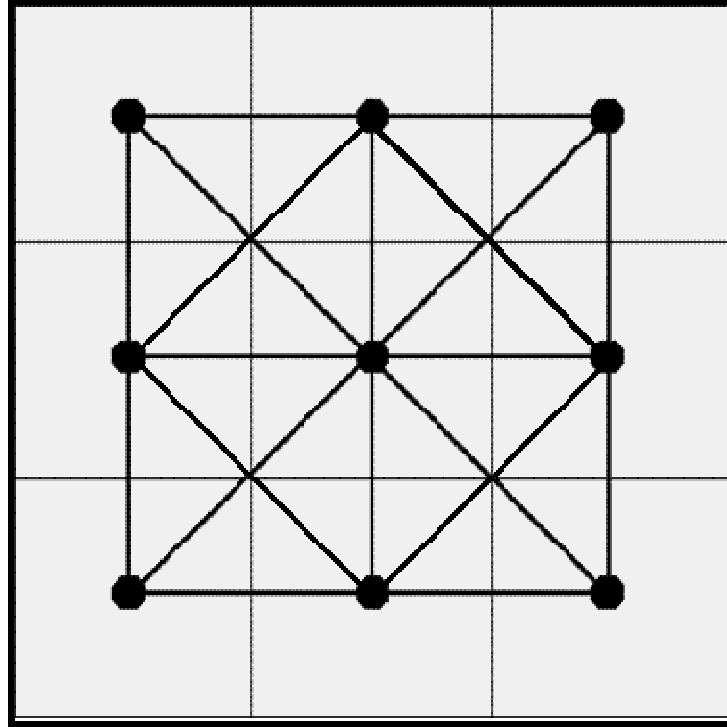


Figure 12. A sample 3x3 terrain layout with 9 nodes and adjacencies to neighboring nodes.

Black points at the center of each square represent nodes and the lines connecting each node to a neighboring node are arcs. A node can have at most 8 adjacent nodes.

If a grid square is a barrier, there is no adjacency to that direction (see Figure 13).

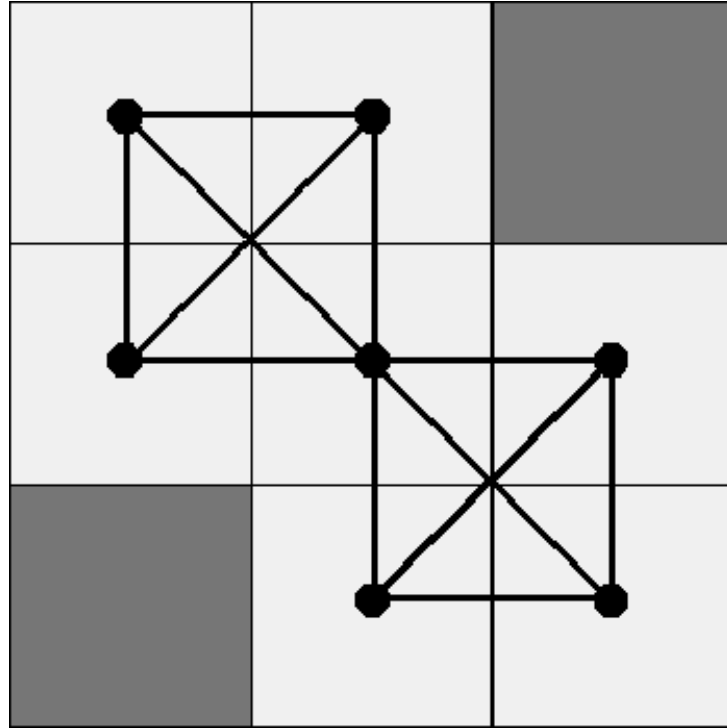


Figure 13. A sample 3x3 terrain layout with 6 nodes and 2 barriers. Black points are nodes, dark-colored squares are barriers. Barriers do not contain nodes and there is no adjacency to them from neighboring squares.

An obstacle is a group of trees, a creek, or a small hill. Grid squares that are not barriers can contain different types of obstacles. A formation cannot cross over a barrier or divide into smaller units to move around a barrier. However, obstacles do not prevent unit movement within the grid squares containing obstacles. They affect the route of the platoon while moving between nodes and the platoon formation can divide into smaller units to pass around obstacles and subsequently form up after passing the obstacles. Figure 14 displays a sample grid terrain layout of the model.

B. FORMATION MOVEMENT IN THE NETWORK

We plan formation movements with a leader-following strategy. First, the platoon leader plans the shortest path from a starting location to a goal location. Next, the leader follows this shortest path in small time-step increments, and his subordinate units determine their successive locations within the formation relative to the leader's location, attempting to maneuver with minimum formation distortion.

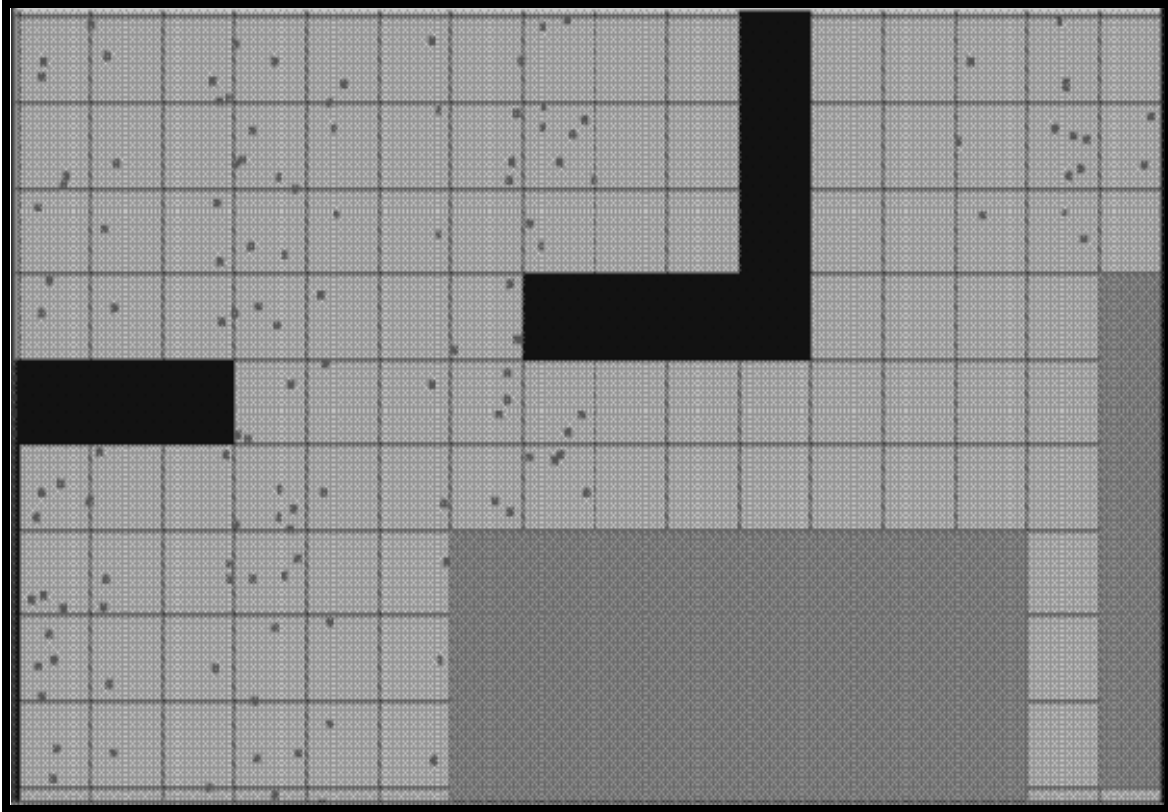


Figure 14. A sample grid terrain layout with barriers and obstacles. The dark-colored areas that cover whole grid squares are barriers, the points within grid squares are obstacles. A formation cannot move into a grid square that is a barrier, but its units can maneuver around obstacles while moving between grid squares.

There are three rules that enable formation movement: *move to a goal location*, *maintain formation* and *avoid barriers and obstacles*.

Move to a goal location and maintain formation rules are the attraction forces for a unit to move to a desired location whereas avoid barriers and obstacles rule represents the repulsion force from barriers and obstacles. Each rule is implemented as a function that generates a vector magnitude and direction for each unit. These rules are taken from the study of Balch and Hybinette [2000].

An instance of formation movement between two nodes is shown in Figure 15.

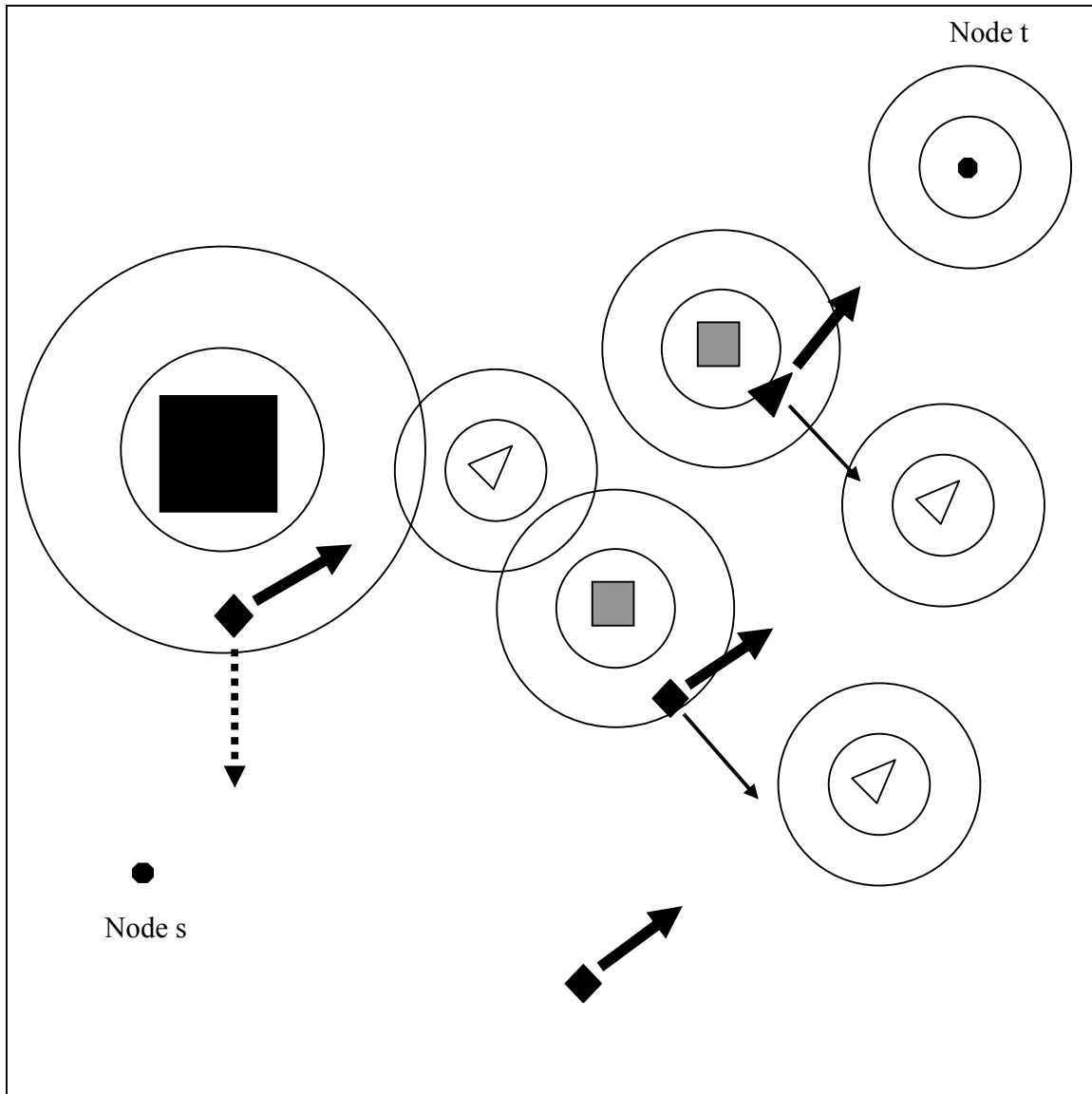


Figure 15. An instance of formation movement of a unit from node s to node t. In Figure 15, the platoon is using a wedge formation. The big darkened square is a barrier and light gray squares are obstacles. The darkened triangle is the platoon leader; white colored triangles with concentric circles are attachment sites for his subordinate units. For the platoon leader there is one attraction force to node t which is shown as a thick arrow, and one repulsion force from the closest obstacle which is shown as a thin arrow. For each subordinate unit, there is one attraction force to one of the attachment sites; one of them has one repulsion force from an obstacle, one of them has one repulsion force from the barrier. The arrows show the directions of the forces on the units, the magnitudes of each force vary with distance from the unit to the obstacle, goal location or attachment site. The concentric circles illustrate the maximum and effective ranges of the objects to generate attraction or repulsion force on the units.

1. Move to a Goal Location

This rule decides the successive locations of the platoon leader while moving along the shortest path calculated by the formation movement shortest path-finding algorithm. A time step simulation is used to move the platoon leader between each node from the starting location until the platoon leader reaches the destination. The goal location for the platoon leader is always the next node within the shortest path. When the leader moves to the next node, he is attracted to the successive node until he reaches the destination.

The move to a goal location function generates a vector magnitude and direction. The direction of this vector is always from the current location of the platoon leader toward the goal location. The parameters used to calculate vector magnitude are shown in Figure 16.

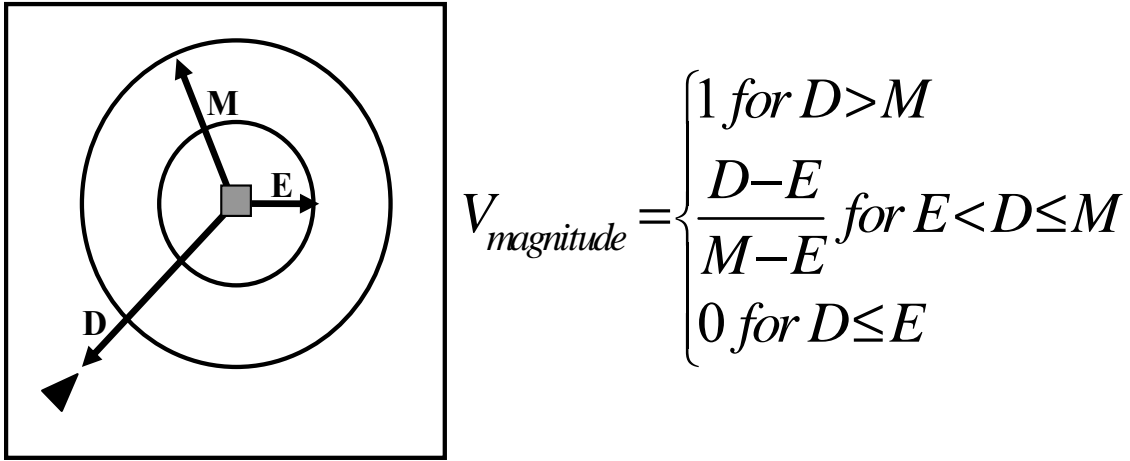


Figure 16. Parameters for move to a goal location and maintain formation rules. D is distance from the unit to the goal location, E is goal location effective range, and M is goal location maximum range. These parameters are used for move to a goal location and maintain formation rules. If a unit is outside the maximum range, it is always attracted to the goal location. If the unit is the platoon leader, then the black point represents a node and the resultant vector is the move to a goal location vector. If the unit is a subordinate unit, the black point represents an attachment site within the movement formation and the resultant vector is the maintain formation vector.

2. Maintain Formation

This rule encourages platoon members to keep formation during their motion through the shortest path. The maintain formation rule does not apply to the platoon leader. Based on the platoon leader's location on the terrain, each squad is attracted to an

attachment site to maintain the formation. Each movement formation provides a different attachment site for each squad.

Figure 17 displays the attachment sites for six different movement formations for a mechanized infantry platoon.

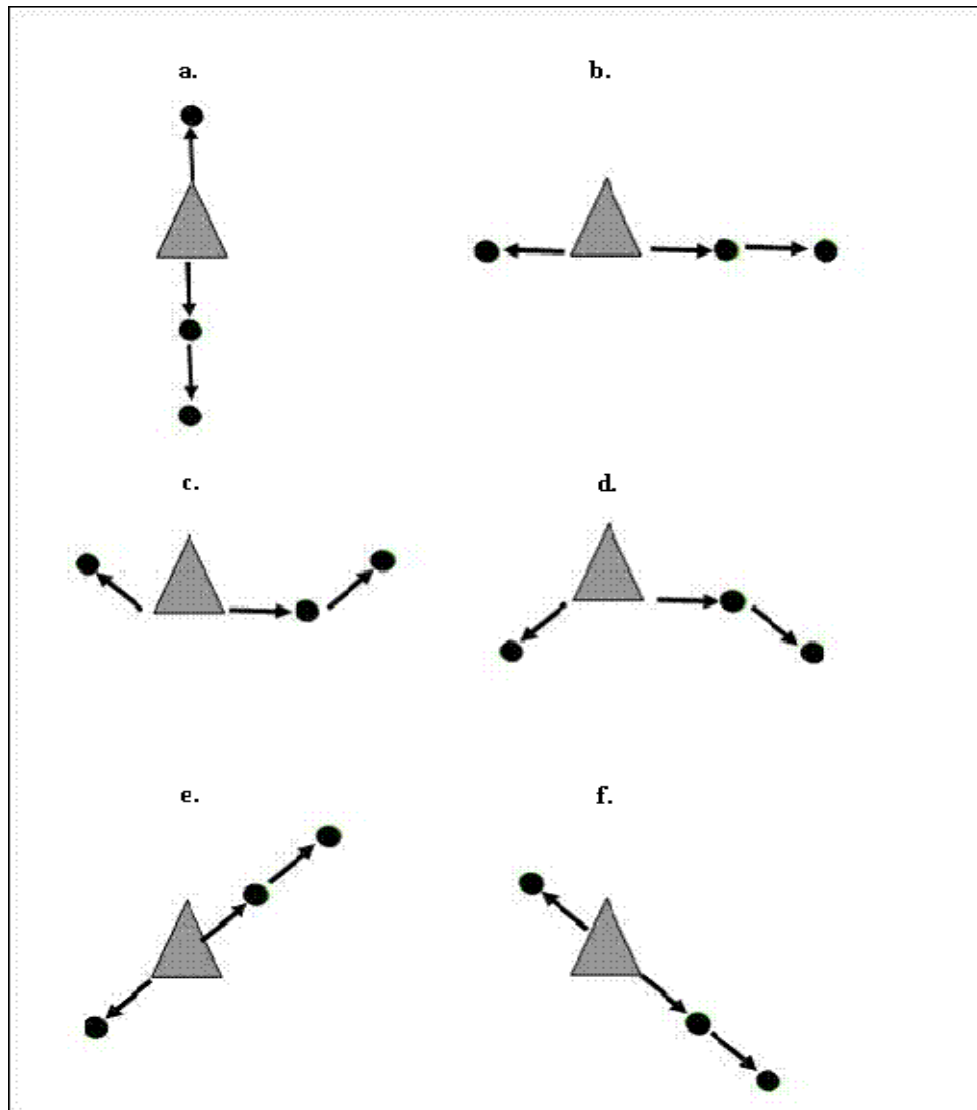


Figure 17. Attachment sites.
a. Column formation b. Line formation c. Vee formation d. Wedge formation e. and f. Echelon formation. The triangle in each formation represents the platoon leader's location and direction. Each squad is attracted to an attachment site (circles) within the formation in order to form into the desired movement formation. Because a mechanized infantry platoon has three squads, each movement formation has exactly 3 attachment sites.

The parameters to calculate maintain formation vector and the formulation are shown in Figure 16.

3. Avoid Barriers and Obstacles

We want to avoid penetrating into barriers or obstacles while navigating over terrain. This rule affects the locations of the platoon leader and his squads, keeping them at a safe distance from barriers and obstacles and enhancing formation integrity.

The avoid barriers and obstacles rule generates a repulsion vector. Parameters used to calculate vector magnitude are shown in Figure 18.

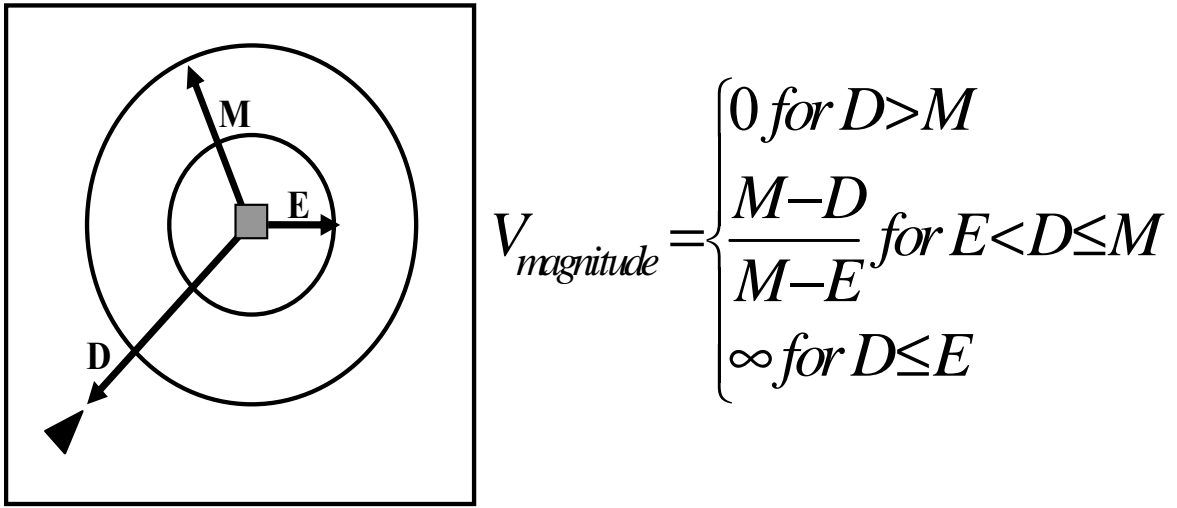


Figure 18. Parameters for avoid obstacles rule.

The square at the center of the inner circle is a barrier or an obstacle. D is distance from the unit to the barrier or obstacle, E is effective range, and M is maximum range. The direction of the resultant vector is always away from the barrier or obstacle through the unit. If a unit is outside the maximum range, the magnitude of the repulsion vector is 0. The formulation to calculate the magnitude of the repulsion vector is also shown. Different maximum and effective ranges of barriers or obstacles may change the unit's path.

4. Formulation

Between each pair of adjacent nodes a time step simulation is used to find the local movement vectors of formation units. These vectors are calculated by a summation of attraction vectors to a goal location or an attachment site that are generated by “move to a goal location” and “maintain formation” rules respectively and repulsion vectors pushing the unit away from barriers and obstacles that are generated separately for each

barrier and obstacle by the “avoid barriers and obstacles” rule. The mathematical formulation to find the movement vector for each unit at each simulation time increment is:

a. Index Use

b	barrier
q	obstacle
i	dimension (x and y)
l	formation leader
u	formation unit
n	goal location for formation leader
s	attachment sites for formation units

b. Data

$V(i,b,l)$	2-dimensional repulsion vector from each barrier on each formation leader
$V(i,b,u)$	2-dimensional repulsion vector from each barrier on each formation unit
$V(i,q,l)$	2-dimensional repulsion vector from each obstacle on each formation leader
$V(i,q,u)$	2-dimensional repulsion vector from each obstacle on each formation unit
$V(i,n,l)$	2-dimensional attraction vector to a goal location for each formation leader
$V(i,s,u)$	2-dimensional attraction vector to an attachment site for each formation unit

Resultant vector for each formation leader:

$$R_{il} = \sum_{\substack{b \\ b | V_{ib l} > 0}} V_{ib l} + \sum_{\substack{q \\ q | V_{iq l} > 0}} V_{iq l} + V_{in l} .$$

Resultant vector for each formation unit:

$$R_{iu} = \sum_{\substack{b \\ b | V_{ib u} > 0}} V_{ib u} + \sum_{\substack{q \\ q | V_{iq u} > 0}} V_{iq u} + V_{is u} .$$

5. A Solution to Computational Complexity

The computational complexity of calculating the resultant vector depends on the number of formation units, barriers and obstacles in the model. The model may have computational difficulties in handling large numbers of these entities.

Table 1 lists the running time to calculate resultant vectors for various numbers of units with 900 obstacles and 875 barriers. Table 2 lists the running time against the same numbers of units with no obstacles and no barriers. Running time is in milliseconds on a Pentium M, 1.6 GHz personal computer. Figures 19 and 20 show the plots of the Table 1 and 2 respectively.

Number of Units	4	8	16	32	64	128	150	200
Running Time	160	310	620	2,423	4,827	9,734	14,070	15,211

Table 1. Running time versus number of units with obstacles and barriers.

8 different numbers of units are tested to find out the effect of number of units on running time. For this set of tests, 900 obstacles and 875 barriers are included in the computation to calculate the resultant vector. With this fixed number of obstacles and barriers, larger numbers of units increase running time of the algorithm. For instance, for 64 units it takes 4,827 milliseconds to calculate the resultant vectors.

Number of Units	4	8	16	32	64	128	150	200
Running Time	0	10	20	30	50	150	191	331

Table 2. Running time versus number of units without obstacles or barriers.

For this set of tests there are no obstacles or barriers included in the computation to calculate the resultant vector. The results show that increase in running time of the algorithm is much slower when the number of obstacles and barriers is 0. For instance, for 64 units it takes 50 milliseconds to calculate the resultant vectors.

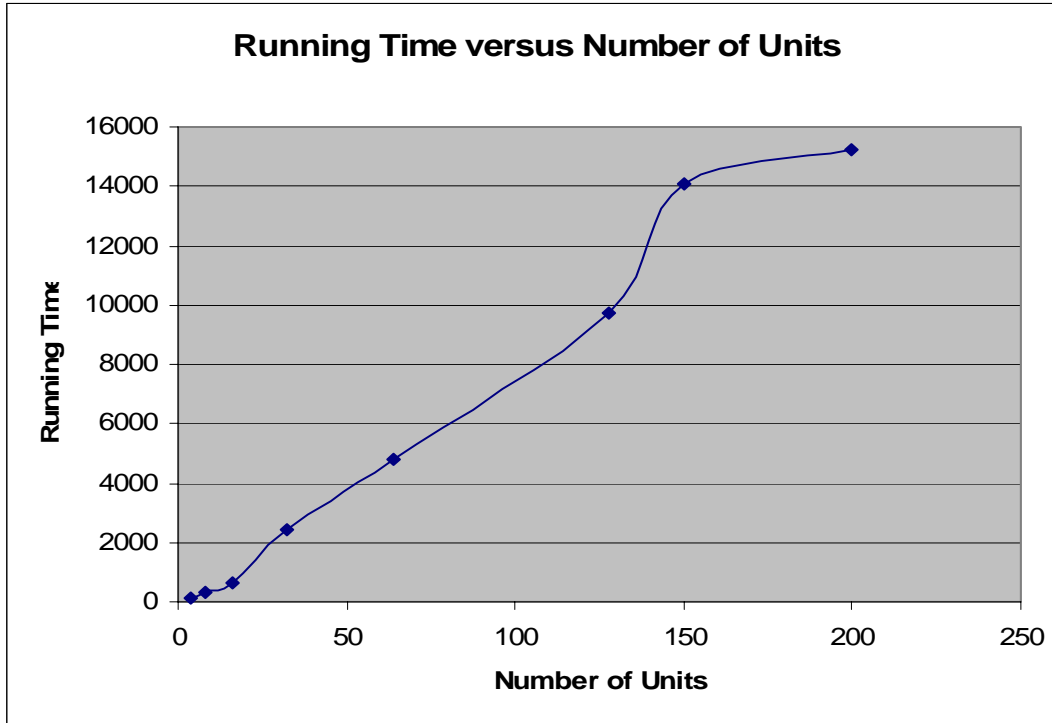


Figure 19. Running time as a function of number of units with obstacles and barriers. This plot shows the effect of number of units on the running time when obstacles and barriers are included in the computation. For instance, for 128 units it takes 9,734 milliseconds to calculate the resultant vectors.

Figures 19 and 20 show how the increase in the number of barriers and obstacles increases running time. To calculate a resultant vector for 200 units with no barriers or obstacles takes 331 milliseconds whereas it takes 15,211 milliseconds with 900 obstacles and 875 barriers. This analysis suggests that the computation problem can be ameliorated if the numbers of barriers and obstacles can be reduced.

When a formation is moving from one grid square to a neighboring square, barriers or obstacles within the surrounding grid squares may create repulsion vectors on the formation units. Because of the maximum and effective ranges of barriers and obstacles they can only generate forces over units when they are within these ranges. Before calculating the resultant vector we already know that most of the barriers and obstacles within the terrain layout do not generate repulsion forces. Creating subsets of barriers and obstacles for each grid square and using these subsets while calculating resultant vector can speed up runtime.

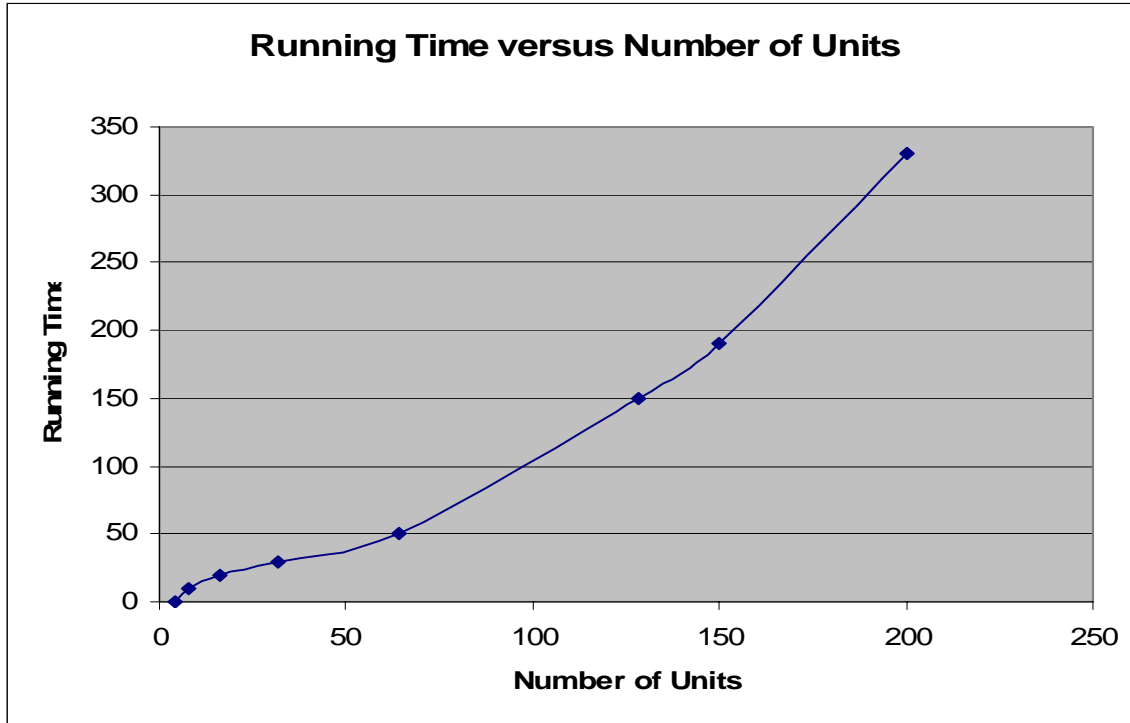


Figure 20. Running time as a function of number of units without obstacles or barriers.

This plot shows the increase in running time when there is no obstacle or barrier included in the computation. Running time still increases with the number of units, but the trend is much smaller than when obstacles and barriers are present. For instance, for 128 units it takes 150 milliseconds to calculate the resultant vectors.

C. FINDING ARC COSTS FOR THE NETWORK

Once the network is established with nodes and adjacent arcs, arc costs are calculated using a time step simulation. The units move in some formation with each formation having different lateral and longitudinal dispersions of its units from the location of the platoon leader. This dispersion factor enforces the concept of maintaining a safe distance from barriers at every point along the shortest path. The time step simulation distinguishes the nodes that have enough barrier clear area for formation movement.

At each time increment in simulation time, the vector sum of forces is calculated for each formation unit. The unit is then moved in the resultant direction before the next time increment. The pseudo code of time step simulation between two nodes is shown in Figure 21.

```

node s=starting location
node t = goal location

for( t=0;t<stop_time; t+= $\Delta t$ ){
    for each unit do;
        calculate attraction vector
        for each barrier do;
            calculate barrier repulsion vector
        end;
        for each obstacle do;
            calculate obstacle repulsion vector
        end;
        calculate resultant vector
        move to resultant vector direction
        calculate total distance from starting location to current location
        arc_cost = distance;
        if distance > upper limit then
            arc_cost =  $\infty$ ;
            t=stop_time;
        end;
        if current location= node t then
            t=stop_time;
        end;
    }
}

```

Figure 21. Time-step simulation to determine travel distance from one node to an adjacent node while following a barrier or obstacle repulsion function.

An arc cost for this network is the distance between two adjacent nodes. Figure 22 shows a time step simulation to calculate the travel distance from node s to node t.

If there are sufficient obstacles influencing the movement of a formation, the avoid obstacles rule can dominate the move to a goal location and maintain formation rules. This could increase the arc costs between two nodes to unreasonable values. To

prevent these unreasonable arc costs in the network, we put an upper limit to the increase of arc costs during the time step simulation. If an arc cost is more than the selected upper limit, then the arc cost is set to infinity (i.e., the adjacency is dropped from the network). Figure 23 shows the relationship between the arc costs and the number of obstacles.

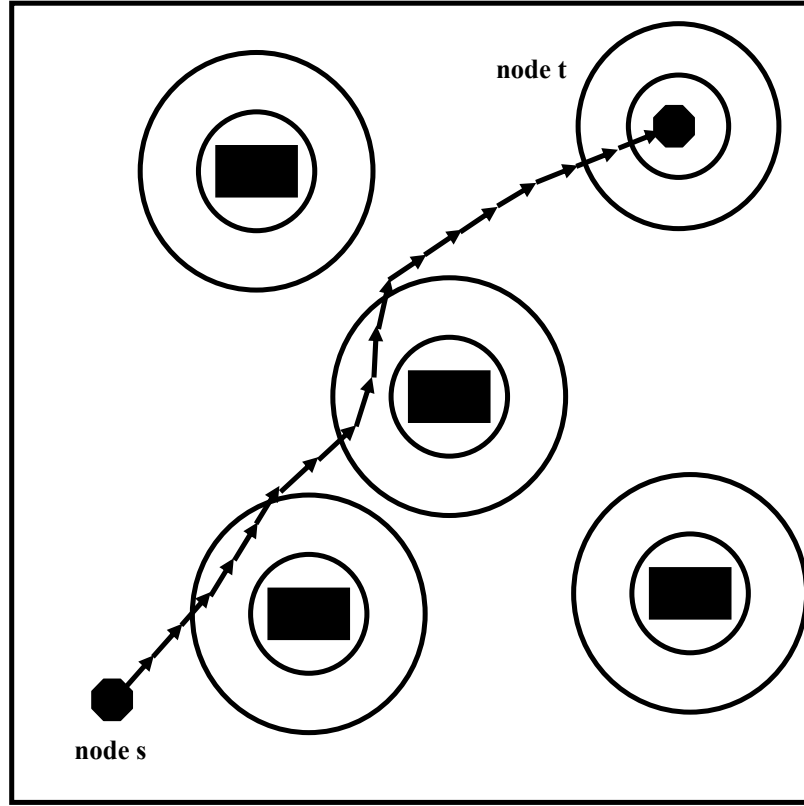


Figure 22. A display to calculate arc costs as a node-to-node time step simulation. The dark rectangles are obstacles that repel the unit navigating from node s to node t. Each obstacle creates a repulsion vector when the unit enters the obstacle's maximum range. The effect of the avoid obstacles function increases the travel distance. The arrows are movement steps at each time increment to the direction of resultant vector.

D. SHORTEST PATH-FINDING ALGORITHM

The shortest path-finding algorithm finds a path that has a desired distance to barriers at every point along the path and is wide enough to permit passage of the formation, while controlling the distortion of relative positions.

The data set for the path-finding algorithm includes node numbers, adjacent node numbers and travel distances between nodes. A sample network with these attributes is shown in Figure 24.

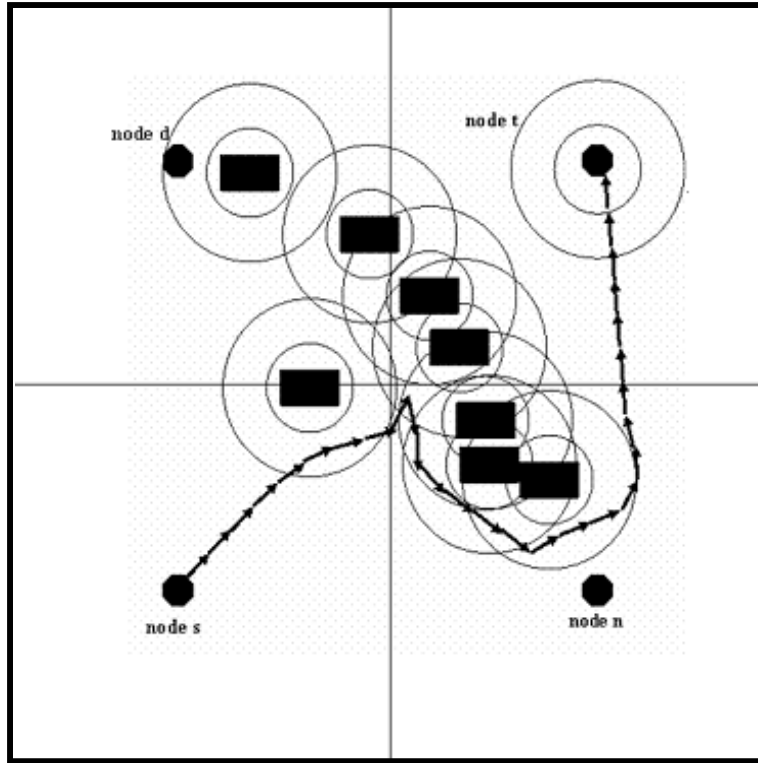


Figure 23. The effect of obstacles on arc cost from adjacent node s to node t. The arrows show the movement direction. The length of each arrow is the magnitude of each movement vector. At each time step increment the formation moves in the arrow direction. When the formation enters the maximum range of an obstacle, a repulsion vector forces movement to be redirected until it is outside of the maximum range. The time step simulation ends when the unit reaches node t and arc cost from node s to node t is the total of the magnitudes of the local movement vectors. In this figure, the arc cost from node s to node t increases substantially. If arc cost(s,t) is greater than the sum of arc cost(s,n) and arc cost(n,t), arc cost(s,t) is set to infinity (i.e., if the direct cost exceeds the rectilinear cost, the adjacency is dropped from the network).

A forward star data structure is used in the Dijkstra shortest path-finding algorithm as shown in Figure 25.

This forward star data structure can be used directly by the 2-heap-Dijkstra shortest path-finding algorithm. The pseudo code for the Dijkstra algorithm is shown in Figure 26 [Ahuja, et al, 1993, p. 115].

If there is a path from node s to node t, the algorithm finds it and it is guaranteed that formation movement can be conducted along this path.

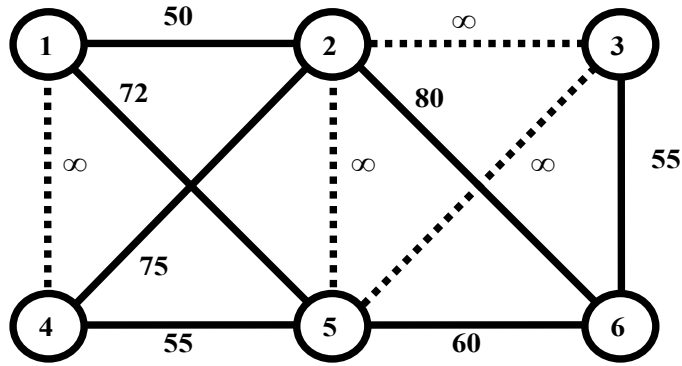


Figure 24. A sample 2x3 terrain network with arc costs. There are 6 nodes in the network. The dashed lines between nodes indicate that during time step simulation, travel distances between these nodes exceeds the upper limit and arc costs for these nodes are set to infinity.

	point	tail	head	cost
1	1	1	2	50
2	3	1	5	72
3	5	2	1	50
4	6	2	4	75
5	8	3	6	55
6	10	4	2	75
7	13	4	5	55
		5	1	72
		5	6	60
		6	2	80
		6	3	55
		6	5	60

Figure 25. Forward star representation of a sample data set (After Ahuja, *et al*, [1993, pp. 35-37]).

Point, tail, head and cost are four arrays to store the data set of the network. For instance, for arc (3, 6), starting node 3 is stored in tail array, ending node 6 is stored in head array, and the arc cost 55 is stored in the cost array. Point array shows array positions of each arc in the network. For arc (3, 6), node numbers and arc cost are stored in the array positions tail(5), head (5) and cost (5).

```

algorithm heap-Dijkstra;
begin
    create-heap(H);
     $d(j) := \infty$  for all  $j \in N$ ;
     $d(s) := 0$  and  $\text{pred}(s) := 0$ ;
    insert(s,H);
    while  $H \neq \emptyset$  do
        begin
            find-min(i,H);
            delete-min(i,H);
            for each  $(i,j) \in A(i)$  do
                begin
                    value: =  $d(i) + c_{ij}$  ;
                    if  $d(j) > \text{value}$  then
                        if  $d(j) = \infty$  then  $d(j) := \text{value}$ ,  $\text{pred}(j) := i$ , insert (j,H)
                        else set  $d(j) := \text{value}$ ,  $\text{pred}(j) := i$ , and decrease-key (value,j,H );
                    end;
                end;
            end;
        end;
    end;

```

Figure 26. Dijkstra's algorithm using a 2-heap (From Ahuja, *et al*, [1993, p. 115]). s is the starting node. $d(j)$ is the cost of a shortest path found from node s to node j . When we run the algorithm, $d(j)$ reveals whether there is a shortest path from node s to node j , and gives its length. The pred array indicates the predecessor node of each node in the shortest path. For instance, $\text{pred}(s)$ is zero as s is the starting node.

IV. MODEL IMPLEMENTATION

We develop a Java application to optimize and animate formation movement of a mechanized infantry platoon over heterogeneous terrain.

A. MOVEMENT SCENARIO

In this scenario, the mechanized infantry platoon is conducting a security patrol over its area of responsibility (AOR) from its assembly area to three successive check points. They first move to check point 1 and must avoid barriers and traverse obstacles. They make similar movements to check point 2 and then 3. Check point 3 is the final destination where the unit establishes a new assembly area. A map of the AOR is shown in Figure 27.

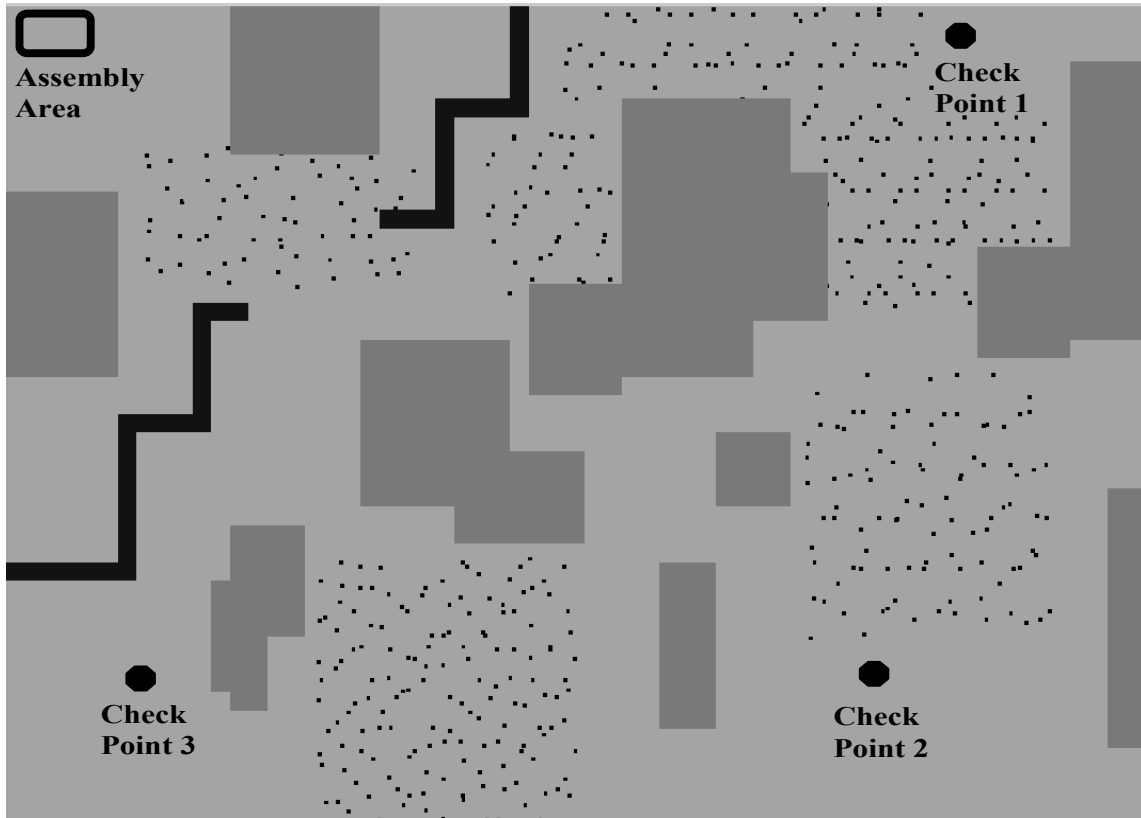


Figure 27. AOR for the mechanized infantry platoon.

This map is from the graphical user interface of the Java application. The starting location is the assembly area, and the platoon will patrol from the assembly area to check points 1, 2 and 3. The two black lines and various sized light gray areas are barriers; the small gray points are obstacles.

Within the AOR, there are different types of barriers and obstacles which complicate the movement of the platoon over the AOR. The platoon will use a wedge formation. To maintain formation integrity while moving over the terrain, the platoon leader must analyze the terrain to plan the movement route of the platoon. During his planning process he will decide:

- How close can the platoon move to a barrier? This is barrier effective range in our model. These attributes provide a clearance around the movement path and guarantee that the platoon does not penetrate any barrier.
- How close can the platoon move to an obstacle? This is obstacle effective range in the model. These attributes ensure that the platoon does not penetrate any obstacle.

B. RUNNING THE MODEL

The maximum and effective ranges of obstacles, barriers, goal locations and attachment sites are adjusted to keep the formation integrity of the unit with minimum distortion. Distance is measured in pixels in the model and each pixel is assumed to be 5 meters over real terrain. The initial parameters used in the model are:

- Barrier maximum range: 70
- Barrier effective range: 35
- Obstacle maximum range: 10
- Obstacle effective range: 5
- Goal location maximum range: 10
- Goal location effective range: 5
- Attachment site maximum range: 6
- Attachment site effective range: 3

These parameters are imported into the model and the shortest path from the starting location to checkpoint 3 is found which is shown in Figure 28.

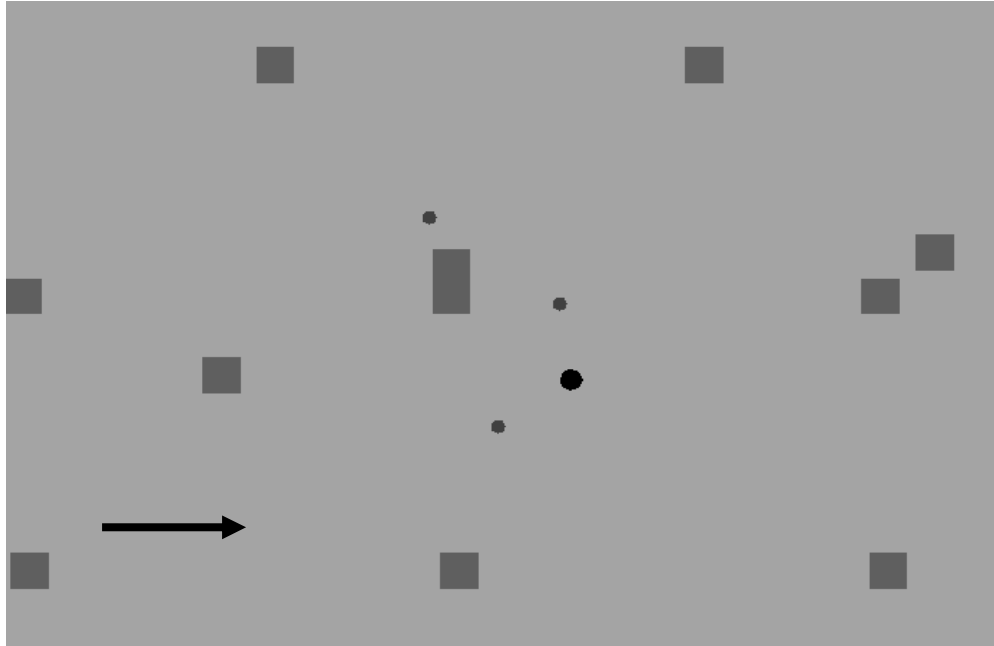


Figure 29. Example of appearance of distorted formation. The big black circle represents the platoon leader; small circles are the squads. The arrow shows the movement direction. The repulsion force of an obstacle on the left wing squad deforms the wedge formation.

The size of the formation and the task may require using various values for attributes. Alternate tuning of the attributes may result in different shortest paths depending on the numbers of barriers and obstacles within the AOR.

Table 3 shows the output of the algorithm for various barrier effective ranges, and a scatter plot of the output is shown in Figure 30.

Barrier Effective Range	Shortest Path Length
35	6,550
50	6,617
70	6,793
80	6,850

Table 3. A comparison of shortest path lengths with various barrier effective ranges. The unit of distance is a pixel (~ 5 meters). Barrier effective range shows if there is a path between two points, this path will at least have a barrier clear area as wide as the effective range along the shortest path. The second column is the total length of the shortest path from the assembly area to the check point 3.

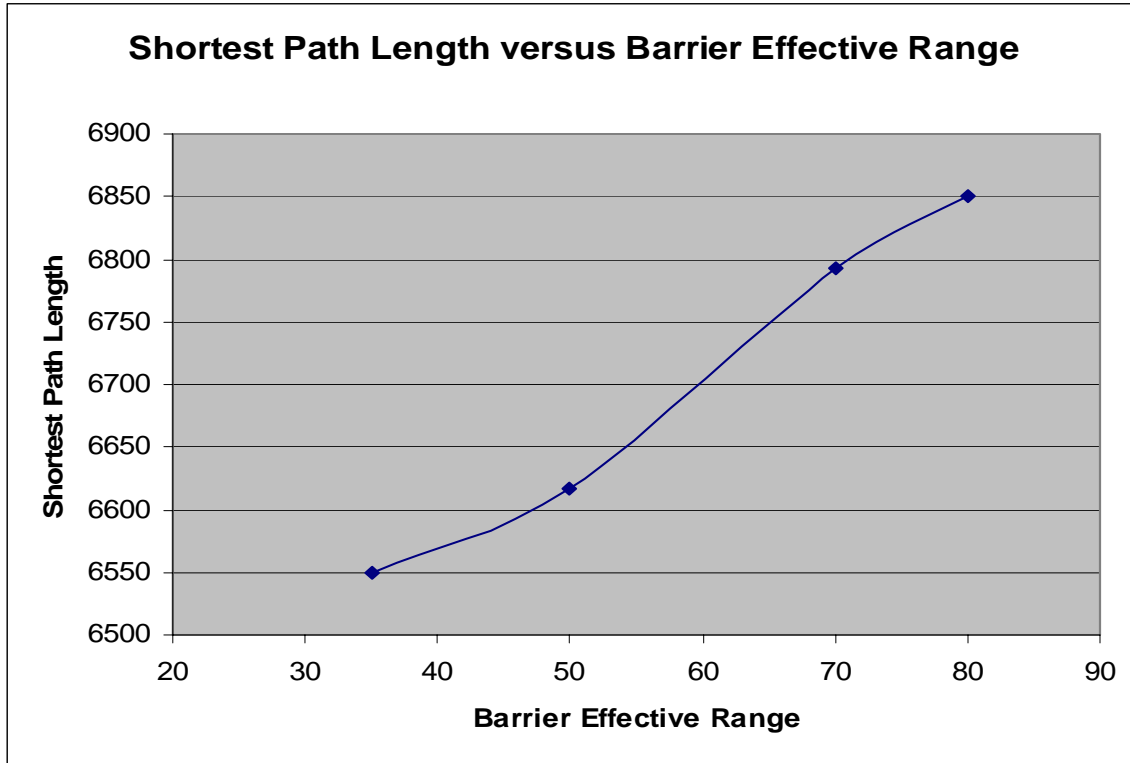


Figure 30. Shortest path length as a function of barrier effective range. This plot indicates that the length of the shortest path increases as the barrier effective range increases. For instance path length from the starting point to check point 3 is 6,617 pixels with a barrier effective range of 50 pixels. This increases to 6,793 pixels if we increase barrier effective range to 70 pixels.

Table 4 shows the output of the algorithm with various obstacle effective ranges, and a scatter plot of the data in Table 4 is shown in Figure 31.

Obstacle Effective Range	Shortest Path Length
5	6,550
6	6,607
7	6,697
8	6,798
9	6,926

Table 4. A comparison of shortest path lengths with various obstacle effective ranges.

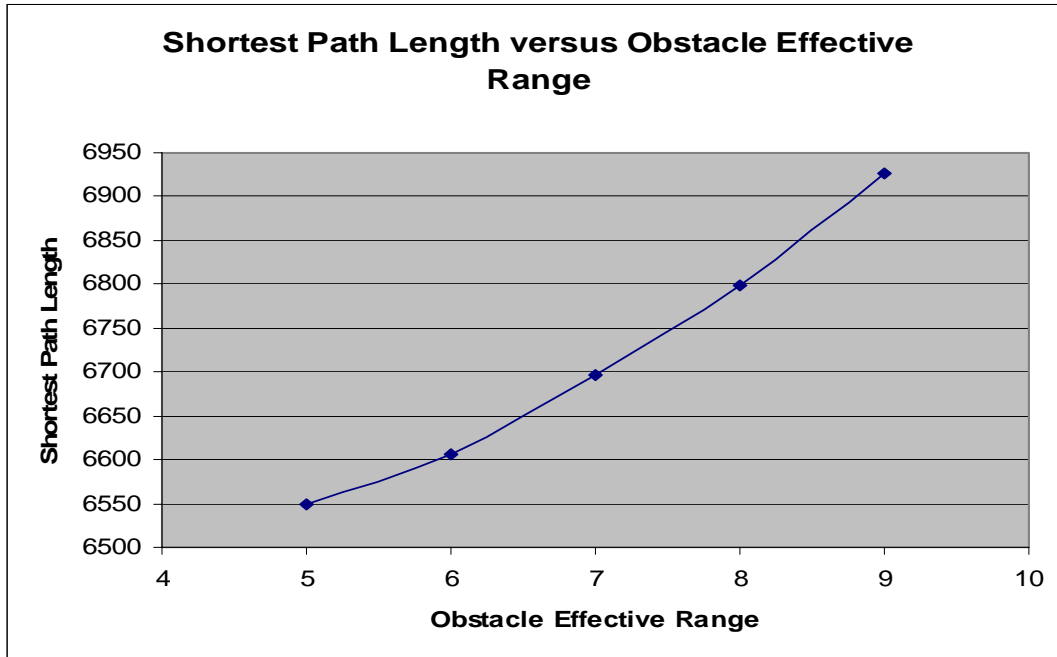


Figure 31. Shortest path length as a function of obstacle effective range. The length of the shortest path increases as we increase the obstacle effective range.

Individual effects of barrier and obstacle effective ranges over the shortest path are illustrated in Figures 30 and 31. Table 5 lists the length of the shortest path with various barrier and obstacle effective ranges. A 3-dimensional plot of this data is shown in Figure 32 to illustrate cooperative effects of barrier and obstacle effective ranges on the shortest path.

		Obstacle Effective Range				
		5	6	7	8	9
Barrier Effective Range	35	6,550	6,607	6,697	6,798	6,926
	50	6,617	6,668	6,788	6,891	7,025
	70	6,793	6,840	6,933	7,100	7,249
	80	6,850	6,899	6,998	7,179	7,345

Table 5. A comparison of shortest path lengths with various barrier and obstacle effective ranges.

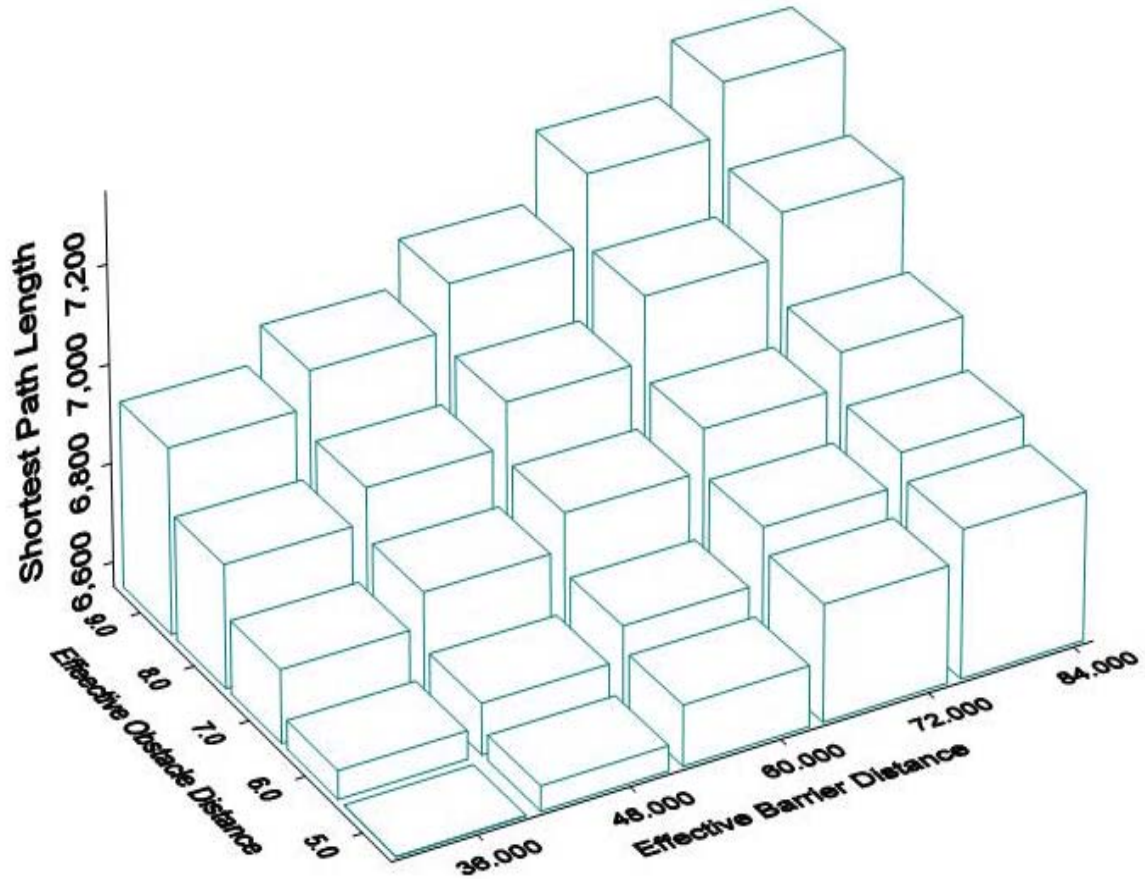


Figure 32. Shortest path length versus barrier effective range and obstacle effective range.
As effective distance increases, the formation is restricted to take longer evasive paths.

C. MAZE PATH-FINDING

In this scenario, the platoon is moving over a maze-like terrain and the goal of the platoon is to move to its objective without penetrating barriers or getting caught in dead ends (cul-de-sacs) that would extend its travel time. The map of the terrain is given in Figure 33.

The output of the application is shown in Figure 34. There is a shortest path connecting s to t .

If there is a path between two points, our model finds the path and animates it as shown in Figure 34. If there is not a path between any given points (Figure 35), the output of the model is shown in Figure 36.

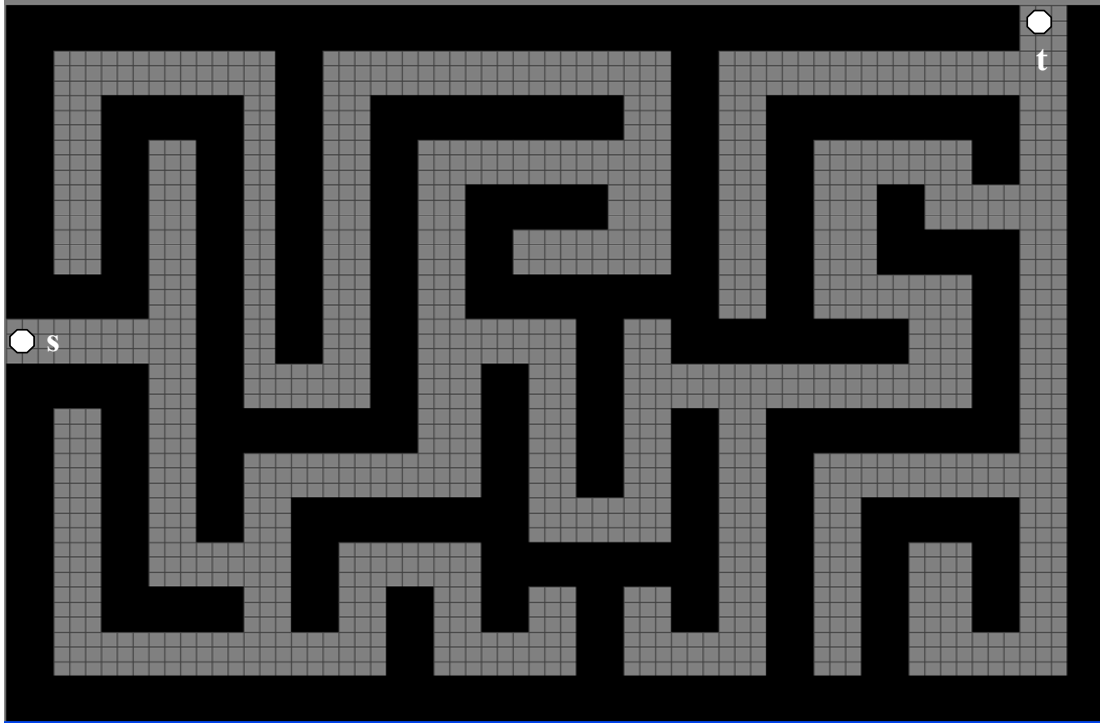


Figure 33. Maze terrain.

Here, s is the starting location and t is the goal location. Dark-colored areas are barriers. The objective is to find the shortest path from s to t .

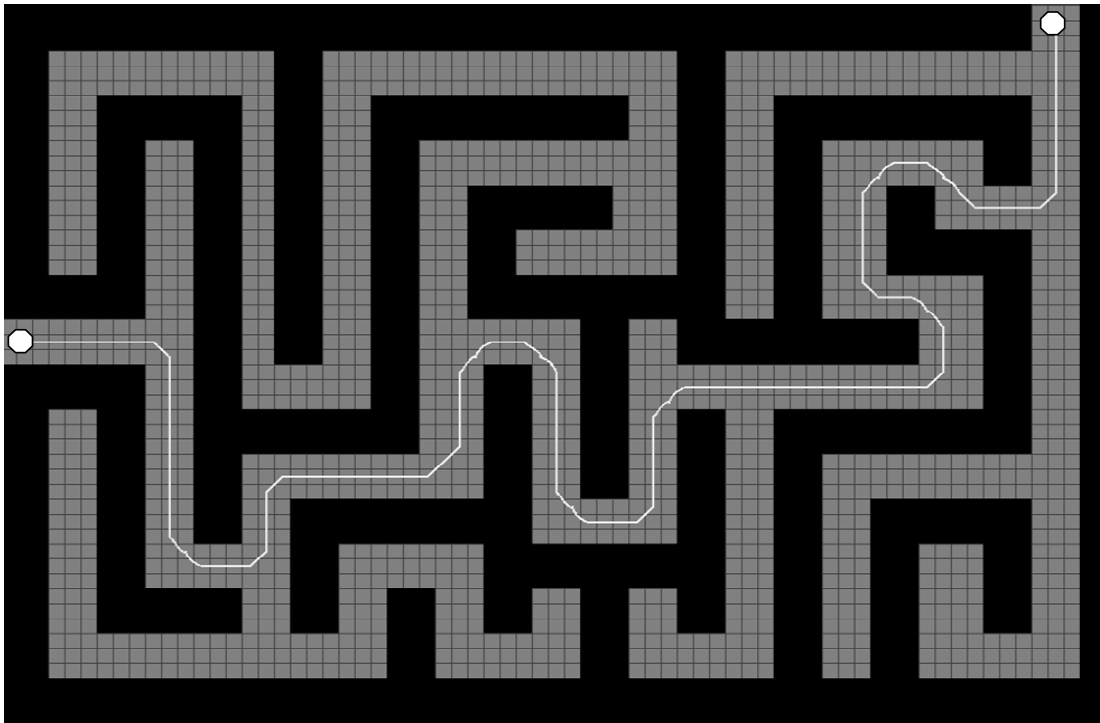


Figure 34. The shortest path from s to t .

The thin line connecting s to t is the shortest path that does not penetrate barriers or get caught in cul-de-sacs.

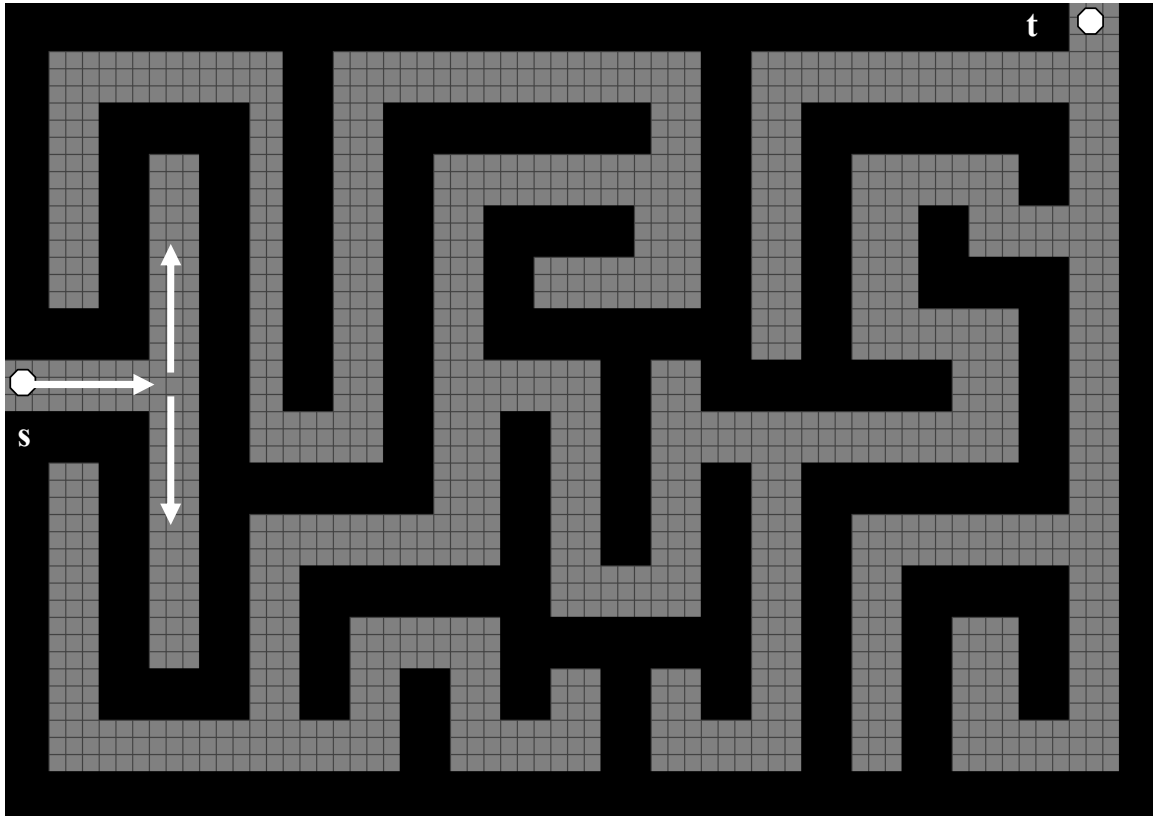


Figure 35. Maze terrain with no path.
The objective is to move from s to t. The light gray arrows show the possible movement directions. In this maze there isn't a path from s to t.

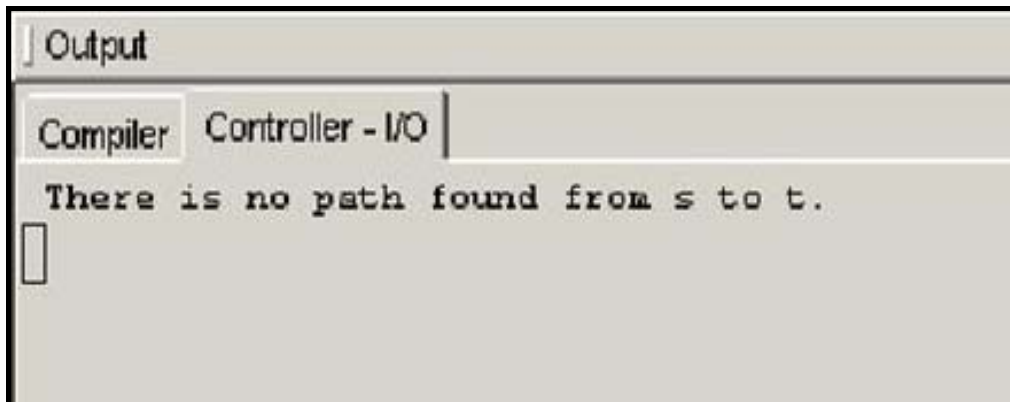


Figure 36. Output window of the Java application.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS

We present a formation movement algorithm that integrates well-accepted tools of robotics control and electronic game design into a path-finding problem.

The time step simulation to calculate the travel distances between adjacent nodes using the motor schemes of robotics, such as the move to a goal location, maintain formation and avoid obstacle schemes, excludes the grid squares that do not allow formation movement from any shortest path.

A distinguishing difference between our method and those used to control robot or video game movement is that we do not plan movement that penetrate barriers. Robot and video game path-finding anticipates continuous, dynamic changes to the features of the terrain. So, myopic methods are used to minimize computational burden. In our case, the terrain is static and we can afford to plan more carefully—omnisciently.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Ahuja, R. K., T. L. Magnanti and J. B. Orlin [1993] *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, Inc., Upper Saddle River, New Jersey.

Aragon, A. R. [2001] *Agent-Based Simulation of a Marine Infantry Squad in an Urban Environment*, Master's Thesis in Computer Science, Naval Postgraduate School, Monterey, California, September.

Balch, T. and M. Hybinette [2000] "Social Potentials for Scalable Multi-Robot Formations," IEEE International Conference on Robotics and Automation (ICRA '00), Vol. 1, pp. 73-80.

Crombie, D [1997] "The Examination and Exploration of Algorithms and Complex Behaviour to Realistically Control Multiple Mobile Robots," Master's Thesis in Engineering and Information Technology, The Australian National University, Canberra ACT, Australia, October.

Java [2005] Sun Microsystems, [<http://java.sun.com>], accessed May 28, 2005.

Kamphuis, A. and M. H. Overmars [2004] "Finding Paths for Coherent Groups using Clearance," Eurographics/ACM SIGGRAPH Symposium on Computer Animation, pp. 19-24.

U.S. Army; Field Manual 7-7 [1985] "The Mechanized Infantry Platoon and Squad (APC)," [<http://globalsecurity.org/military/library/policy/army/fm/7-7/Ch4.htm#s4>], Accessed April 2005.

U.S. Army; Field Manual 7-8 [1992] "Infantry Rifle Platoon and Squad," [<http://globalsecurity.org/military/library/policy/army/fm/7-8/ch1.htm#s2>], Accessed April 2005.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California
3. Kara Kuvvetleri Komutanlığı
Personel Daire Başkanlığı
Bakanlıklar
Ankara, Turkey
4. Kara Kuvvetleri Komutanlığı
Kara Harp Okulu Kütüphanesi
Bakanlıklar
Ankara, Turkey
5. Distinguished Professor Gerald G. Brown
Naval Postgraduate School
Monterey, California
6. Ellen F. Roland
Rolands & Associates Corporation
Monterey, California
7. Fatih Cesur
Piyade Okul Komutanlığı
Tuzla
Istanbul, Turkey